Efficient Resource Optimization in Cloud-Fog Computing Environment Using Soft Computing Techniques

Synopsis

Submitted in the partial fulfilment for the requirement of

Doctor of Philosophy

In

Computer Science

Submitted by

SAIYAM VARSHNEY

Under the supervision of

Prof. Gur Mauj Saran Srivastava

Department of Physics and Computer Science, Faculty of Science
DAYALBAGH EDUCATIONAL INSTITUTE (DEEMED UNIVERSITY)
DAYALBAGH, AGRA-282005

2022

1. Introduction

In this era of digitalization, the advancement of modern technologies have not only captured and recorded a large number of data sets in various fields but on the other hand, many difficulties in storage and processing have been noted. As a solution to address the above problem comes the new concept, which is based on Distributed Computing using the Internet. At the same time, diversity in the size and format of the data is constantly increasing, resulting in the new challenges of presenting it as additional versatile processing tools and platforms, so that useful information can be discovered from anywhere [1].

Most of the new big data solutions are built on cloud computing technology, which has led to the development of various computation services to accommodate the huge increase in volume and velocity of big data.

Cloud computing technologies proved to be very efficient in storing large amounts of data and providing access to it from anywhere in the globe. The nature of cloud computing environments is meant to provide significant benefits, such as cost-effective and reliable, scalable, and computing large datasets on a network across the Internet or all regions to enhance the performance of analytical solutions [2].

1.1. Fog Computing

Fog computing is defined as a distributed computing paradigm that fundamentally extends the services provided by the cloud to the edge of the network [3]. Cisco defines, fog computing is considered as “an extension of the cloud computing paradigm from the core of the network to the edge of the network”. It facilitates the computation, storage, and networking between the end devices and the traditional cloud servers [4].

Fog computing, also known as edge computing, is planned for distributed computing where many "peripheral" devices connect to a cloud [5].

1.1.1. Definition

The term fog computing originally coined by ***Cisco refers to an alternative to cloud computing. This approach seizes upon the dual problem of the proliferation of***

computing devices and the opportunity presented by the data those devices generate by locating certain resources and transactions at the edge of a network[6].

1.1.2. The architecture of Fog Computing

Fog computing is a new computational paradigm, which extends conventional cloud computing and services to the edge of the network [6]. It provides the computation, communication, controlling, storage, and service capabilities at the edge of the network. The decentralized platform differs from other traditional computational models in architecture. In this section, we will discover the architecture and features of fog computing and comparison with cloud computing and edge computing [7].

The Fog computing architecture is composed of the following three layers:

Layer 1: Edge

The edge means that it is closest to the end-user and the physical environment. This includes various IoT devices, for example- sensors, mobile phones, and other such types of devices. These devices are widely distributed geographically in general and are responsible for sensing the feature data of physical objects or events and transmitting these sensory data to the upper layer for processing and storage [8].

Layer 2: Fog

The Fog is located at the edge of the network. The fog computing layer consists of several fog nodes, including routers, gateways, switches, connections, stations, fog specifications, etc. These fog nodes are widely spread around cloud and end devices, e.g., cafes, shopping malls, bus terminals, roads, parks, etc. It can be stationary or mobile in a moving carrier in a steady state. Different types of terminals can be easily connected to fog nodes to receive different services. The data received can be temporarily calculated, transmitted, and stored. The fog layer can also be used to perform real-time analysis and latency-sensitive applications. Also, Fog nodes are connected to the cloud data center by an IP core network, which ensures interaction and collaboration with the cloud to enhance processing and storage capabilities [6]-[8].

Layer 3: Cloud

The cloud consists of multiple high-performance servers and storage devices and provides various application services, such as smart homes, smart transportation, smart factory, etc. It has powerful computing and storage capabilities to support extensive computation analysis and permanent storage of an enormous amount of data [9][10].

1.2. Fog Computing Essential Characteristics

The following six characteristics are essential in distinguishing fog computing from other computing paradigms. However, a smart end-device or IoT user is not required to make use of all characteristics when consuming a fog computing service [12]-

i. **Contextual location awareness and low latency-** Fog computing offers the lowest-possible latency due to the fog nodes' awareness of their logical location in the context of the entire system and of the latency costs for communicating with other nodes. The origins of fog computing can be traced to early proposals supporting endpoints with rich services at the edge of the network, including applications with low latency requirements. Because fog nodes are often co-located with the smart end devices, analysis and response to data generated by these devices are much quicker than from a centralized cloud service or data center.

ii. **Geographical distribution -** In sharp contrast to the more centralized cloud, the services and applications targeted by fog computing demand widely, but geographically identifiable, distributed deployments. For instance, fog computing will play an active role in delivering high-quality streaming services to moving vehicles, through proxies and access points geographically positioned along highways and tracks.

iii. **Heterogeneity-** Fog computing supports the collection and processing of data of different form factors acquired through multiple types of network communication capabilities.

iv. **Interoperability and federation -** Seamless support of certain services (real-time streaming services is a good example) require the cooperation of different providers.

Hence, fog computing components must be able to interoperate, and services must be federated across domains.

v. **Real-time interactions-** Fog computing applications involve real-time interactions rather than batch processing.

vi. **Scalability and agility of federated, fog-node clusters-** Fog computing is adaptive, at cluster or cluster-of-clusters level, supporting elastic compute, resource pooling, data-load changes, and network condition variations, to list a few of the supported adaptive functions.

1.3. The technological paradigm shift from the cloud to fog computing

Cloud computing and fog computing provide storage, application, and data to both end-users. However, Fog computing is close to end-users and has a wide geographic distribution [8]-[10].

'Cloud computing is the practice of using data to access, manage, and process networks of remote servers hosted on the Internet, rather than a local server or personal computer [24].

The fact that cloud computing is not sufficient to manage the large amount of data produced by the increasing amount of Internet-of-Things (IoT) devices is making fog computing popular. The number of devices was 20.35 billion in 2017, has reached 30.73 billion in 2020, and is expected to be 80 billion in 2025.

Also known as edge computing or fogging, it makes the operation of fog calculations between end devices and cloud data centers easier to run computing and storage services. While edge computing is commonly referred to as a location for services, fog computing implies a distribution of the communication, computation, storage resources, and services on or close to devices and systems in the control of end-users[12][13]. Fog computing provides a medium-weight and intermediate level of computing power [28]. Instead of alternatives, fog computing often serves as a complement to cloud computing [14].

1.4. Fog Node Architectural Service Models

Fog computing is similar to the traditional cloud computing model and offers implementations of the architecture in multiple layers of the network's topology. Similar to

the cloud computing service models defined in NIST Special Publication (SP) 500-325[10], the following types of service models can be implemented:

1.4.1. Software as a Service (SaaS) - The capability provided to the fog service customer is to use the fog provider's applications running on a cluster of federated fog nodes managed by the provider. This type of service is similar to the cloud computing Software as a Service (SaaS) and implies that the end-device or smart thing accesses the fog node's applications through a thin client interface or a program interface. The end-user does not manage or control the underlying fog node's infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

1.4.2. Platform as a Service (PaaS) -The capability provided to the fog service customer is also similar to the cloud computing Platform as a Service (PaaS) and allows deployment onto the platforms of federated fog nodes forming a cluster, of customer-created or acquired applications created using programming languages, libraries, services, and tools supported by the fog service provider. The fog service customer does not manage or control the underlying fog platform(s) and infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

1.4.3. Infrastructure as a Service (IaaS) -The capability provided to the fog service customer is to provision processing, storage, networks, and other fundamental computing resources leveraging the infrastructure of the fog nodes forming a federated cluster. Similar to cloud computing Infrastructure as a Service (IaaS) services, the customer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying infrastructure of the fog nodes cluster but has control over operating systems, storage, deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

1.5. Fog Node Deployment Models

Since fog computing is identified and defined as an extension of the traditional cloud-based

computing model, the following deployment models are also supported:

1.5.1. Private fog node. A private fog node that is offered for exclusive use by a single organization with multiple consumers (e.g. business units). It may be owned, managed, and operated by an organization, third party, or a combination of some of them, and it may exist on or off-premises.

1.5.2. Community fog node. A community fog node that is offered for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations.) It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off-premises.

1.5.3. Public fog node. A public fog node that is offered for open use by the general public. It may be owned, managed, and operated by a business, academic, government organization, or some combination of them. It exists on the premises of the fog provider.

1.5.4. Hybrid fog node. A hybrid fog node that is a composition of two or more distinct fog nodes (private, community, or public) that maintain unique entities, but is bound together by standardized or proprietary technology that enables data and application portability (e.g., fog bursting for load balancing between these fog nodes.)

2. Underlying Technology

2.1. Fog Virtualization: Network virtualization is considered to be a very important enabling technology in the context of fog computing. It refers to intangible network resources that were traditionally distributed across the hardware to software. It can combine multiple physical networks into one virtual, software-based network, or it can divide a physical network into separate, independent virtual networks. It separates resources and services from the physical delivery environment. This helps to hide computing resource characteristics from users. Network virtualization means "something that is not real" but offers the features of a real.

Virtualization has three main characteristics which make it more useful for fog computing – **Isolation, Partitioning, and Encapsulation.**

2.2. **QoS:** Quality of service (QoS) is the use of mechanisms or technologies to control traffic and ensure the performance of critical applications. It enables organizations to adjust their overall **network traffic** by prioritizing specific high-performance applications.

2.3. **Load Balancing:** Load Balancing has emerged out as a core technological aspect in the fog computing environment. It is useful in even distribution of the processing tasks among the various available clouds, for the efficient utilization of the computing resources which in turn increases the performance and availability of the cloud.

3. Fog Computing Issues and Challenges

Researchers identify many issues and challenges which are required to be addressed properly in an effective way[17]:

3.1. **Latency Constraints**- Fog nodes support the same basic features that clouds can perform, i.e., different computing tasks closer to the end-user, where the latency-sensitive application benefits most from its stringent requirements.

3.2. **Network Bandwidth constraints**- The fog paradigm offers the ability to carry out data processing tasks closer to the network edge. As a result, the amount of raw data sent to the cloud is reduced. Therefore, performing data analysis in fog devices reduces the latency in response while filtered data is sent to the cloud for long-term storage.

3.3. **Resource-constrained devices** - For resource-constrained edge devices such as smartphones and sensors, fog computing can perform computational tasks. The energy consumption and life-cycle costs are reduced by discharging parts of the application from such restricted devices to nearby fog nodes.

3.4. **Increased availability** - Fog Computing supports autonomous operation without depending on the cloud's network connectivity. As a result, an application is increasingly available effectively and efficiently.

3.5. **Better security and privacy-** In the fog paradigm users have the ability to control the collected data. Fog devices reduce the need to transfer private data toward the cloud while such sensitive data stays and is processed locally. Hence, fog devices increase security as

well, being able to perform a wide range of security functions, manage and update the security credentials of constrained devices and monitor the security status of nearby devices. Also, data integrity and privacy are ensured even more when data has to travel shorter network distances to reach the computation node.

4. Literature Review

Many researchers and scientists have proposed different techniques in the area of load balancing and job scheduling in the cloud fog computing environment. A few important load balancing techniques relevant to our research work are discussed below:

Fog computing is a classification that overlays edge devices and cloud computing. According to [21]Hoang et al. in fog environments, routers are potential robust servers that could provide sources for fog services.

In [22] Mtshali et al. proposed an application scheduling technique based on virtualization techniques to find an efficient algorithm that can optimize the energy consumption and average delay of real-time applications in a fog computing network. This is achieved by implementing four task scheduling policies in the Fog node scheduler to assess their performance and efficiency.

In [23] Verma et al. introduced the different architecture, a new real-time efficient scheduling (RTES) policy for load balancing in a fog computing environment, which allows real-time tasks to be completed within the time frame, increasing throughput and network usage, ensuring current time and maintains data consistency with reduced complexity to meet end-users.

In [24] Hoang et al. consider a fog-based region and cloud (FBRC) in which requests are handled locally by not only one region but multiple regions when additional resources are required. Thus an efficient task scheduling mechanism is thus required to reduce the completion time of tasks and the level of satisfaction. To deal with the complexity of scheduling tasks, a heuristic-based algorithm is proposed based on their construction and validated by extensive simulations.

In [25] Hussain et al. introduced the Vehicular Fog Computing(VFC) architecture for a multi-objective optimization (MOO) analytical framework to reduce response time and power

consumption. The vision of VFC has been studied as a complement to cloud computing and is an essential component of the Intelligent Transportation System (ITS) infrastructure. The authors used the concepts of probability and queuing theory to present a model for handling the workload of vehicles in different layers of the VFC architecture. By solving the MOO model with a Modified Differential Evolution (MDE) algorithm, they demonstrated how beneficial the delay-reducing fog-fluctuating policy can be for mission-critical ITS applications.

In [26] Liu et al. propose to optimize the energy consumption and performance delays of mobile devices (MD) using the offloading method. They derived analytical results of objective metrics by considering three types of queuing models in MDS, explicit views of wireless channels, fog, and central cloud. From the results, they used the integer programming model (IPMs) algorithm. Evaluation of results is higher than their plan at the current level. However, the authors did not consider evaluating only scheduling methods based on fog, as the fog was argued to be efficient compared to the cloud. Furthermore, they did not consider performance metrics such as execution time and network usage.

In [27] Natesha et al. propose a first-fit decreasing greedy approach to address the challenge of selecting a fog node that is suitable for hosting IOR applications. This placement problem aims to reduce application latency and energy consumption for the efficient use of fog node resources. Validation of the algorithm was evaluated with two placement approaches; either application are deployed in the fog-cloud interplay or only in the cloud. However, this method was tested to be efficient only on fog vs. cloud. They did not consider the performance evaluation of the deployment algorithm in fog itself.

In [28] Kabirzadeh et al. and [29] Tani et al. present various existing scheduling algorithms and propose hyper-heuristic algorithms for evaluation in fog computing. These several scheduling algorithms include FCFS (First Come First Saved), PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization Algorithm), SA (Simulated Annealing Algorithm), and GA (Genetic Algorithm), etc. However, not all algorithms that contribute to FCFS have been evaluated. Therefore, we intend to investigate classical scheduling algorithms that have been implemented in centralized cloud architecture.

In [30] Singh et al.proposed a combination of allocation and selection techniques with an optimal parameter stack for fully timed decisions for fog computing resource allocation methods. It mainly focused on reducing workload by implementing faster task processing, while also incorporating subgroup-oriented fixation on available resources. Cost is believed to be improved by improving the operating length ratio to improve user length, which ultimately reduces customer churn, and can effectively increase operating revenue. Monitoring the occurrence of failure also has an important role in avoiding computing resources with a high probability of failure. The proposed model is learned to reduce queue size by allocating resources effectively, resulting in quick completion of user workflows.

In [31] Ghobaei-Arani et al. studied different resource management techniques of fog computing and provide a systematic review in the form of taxonomy. The authors emphasize different aspects of resource management, i.e. load balancing, resource scheduling and allocation, task allocation, resource provisioning, and task offloading. The proposed resource management approaches are compared by considering various factors: QoS metrics, the tool used, case studies, and the technique used. Their advantages and drawbacks also compare these approaches.

In [32] Agarwal et al. performed a practical design and algorithm for provisioning the resources using a virtualization method. A capable resource allocation design and algorithm was introduced and executed on a cloud analyst tool to examine the review of the suggested procedure in the fog Background was proposed by him. On implementing the recommended approach, we notice that the prescribed method can be assigned resources in an optimized form. This method is more attractive than others when overall reply time, cost for data transfer, and utilization of bandwidth in the fog computing environment is considered.

In [33] Awaisi et al. present a fog-based efficient architecture for IoT-based healthcare systems that include the idea of a virtual machine (VM) partitioning in fog nodes to smoothly run the body sensor network and medical IoT devices data. The proposed fog-based architecture overcomes the cloud computing issues in healthcare systems and guarantees smooth and efficient processing of patient data in less time with minimum network bandwidth usage.

In [34] Basset et al. proposed a modification marine predators algorithm (MMPA) for tackling the task scheduling in Fog computing (TSFC) to improve the QOSs required by users.

In [35] Deve et al. proposed a decentralized trust model to maintain the reputation of publicly available fog nodes. The proposed trust model is designed using public Ethereum blockchain and smart contract technologies for enabling decentralized trustworthy service provisioning between IoT devices and public fog nodes. The proposed approach is tested and evaluated in terms of security, performance, and cost.

In [36] Rafique et al. proposed a novel bioinspired hybrid algorithm (NBIHA) which is a hybrid of modified particle swarm optimization (MPSO) and modified cat swarm optimization (MCSO). This proposed work is to reduce the average response time and optimize resource utilization by efficiently scheduling the tasks and managing available fog resources.

5. Soft Computing Techniques

In the real world, we cannot solve many issues scientifically or we can claim it technically, but due to large-scale resources and computational time requirements, the challenges are very huge. Bio-natural methods sometimes prove to be efficient and effective for such problems. A near-optimal solution is sometimes more beneficial in most practical problems. Such biological inspired methods are called advanced computing techniques. Modern computing differs from traditional hard computing because it is tolerant of uncertainty, inefficiency, low solution cost, approximation, and robustness.

The basic technical subjects of soft computing are Genetic Algorithm (GA), Probabilistic Reasoning (PR), Neural Computing (NN), and Evolutionary Computing (EC): Modern automatic control systems in manufacturing or process industries face increasing demands regarding performance, product quality, energy, and material consumption. On the one hand, these demands have led to a stronger integration of the processing stages and subsystems, on the other hand to plant-wide hierarchical information and control systems with various man-machine interfaces integrating control, scheduling, supervision, fault detection, and diagnosis. Consequently, the processes to be controlled as well as the control systems have become far more complex. In addition, nonlinear behavior and time-varying characteristics of the processes

have to be taken into account. Furthermore, especially on the higher level of automation systems, information is partly incomplete, vague, and imprecise. For automatic control in consumer products, transportation systems, machines, and so on, analog trends can be observed.

5.1. Fuzzy System

Fuzzy System was introduced by Lotfi A. Zadeh in 1965. The main inspiration behind the introduction of the fuzzy system was the necessity for modeling real-world phenomena, which are inherently vague and ambiguous. Human knowledge about complex problems can be successfully represented using the imprecise terms of natural language. The concept of a system provides formal tools for mathematical representation and efficient processing of such information.

Fuzzy logic has been used in several IoT applications such as smart homes, connected vehicles, real-time analytics, industrial automation, Healthcare, and traffic control system.

5.2. Multi-Objective Optimization

Many optimization problems naturally have several or multiple objectives which need to be satisfied but these objectives may be conflicting in nature, so in order to find the solution of such problem, they may be treated as there is only one such objective while the other may be handled as constraints [20].

Evolutionary multi-Objective optimization can be classified as follows:

A. **Non-Pareto techniques** – In this approach, the multiple objectives can be converted into a single one by using some weight function or using one for the optimization while the other transforms into a constraint. But such a method may result in a huge loss of significant trade-off and unable to find a global optimum.

B. **Pareto-based techniques** – In this approach, a solution that presents optimal trade-offs is selected. The main goal is to find non-dominated global solutions. Such a set of non-dominated solutions are known as the Pareto-Optimal set which is named after Vilfredo Pareto [20].

6. Motivation

Nowadays a large number of end-users are storing their data on the cloud so that they can access

it from anywhere and at any time. But on the other side, users are also facing the problem of latency due to traffic on the cloud. Therefore, to avoid this problem the Fog computing environment was introduced which basically improves latency issues due to the cloud [18].

Fog computing now evolved by adopting existing technologies aimed at low cost and ease of resource management. Virtualization is an underlying technology in fog-cloud computing that enables operation and also maximizes the use of computing resources.

1. Load balancing is also one of the major issues. Load sharing problems occur when multiple user tasks are trying to access the same cloud, while multiple requests to the cloud from other edge/IoT devices become unusable. This condition is known as distributed load imbalance system. The problem can be resolved by scheduling requests or tasks before it arrives in the cloud layer.

   The increased need is to understand precedent load distribution techniques in fog computing. By efficiently balancing the load provided to the fog layer, resource utilization can be enhanced, which will help to reduce the wastage of resources at the fog layer.
2. Fog computing has different layers and each of its layers has latency and computation. Whereas applications need to be allocated to the layers of fog to achieve a better quality of services (QoS) [21].
3. The energy wastage by fog nodes is also increasing, which may cause failure in the fog network. So this environment needs optimal support to reduce the energy consumption in fog nodes.

7. Research Gaps in Literature

Research gaps are identified on the basis of published data in the relevant field. It mentions several challenges regarding cloud-fog architectures such as latency, resource constraint, mobility, security, and privacy which are yet to be explored further.

In literature, resource constraint, load balancing, heterogeneity, unpredictable arrival rate, and a vast number of tasks to be completed make resource and task scheduling a complex issue from a cognitive perspective. The study of the cloud-fog platform from a computational perspective using efficient resource optimization and task scheduling is not explored much.

8. Proposed work

Efficient resource management in fog computing can be considered a prerequisite for maximum or efficient utilization of computing resources to cloud computing environments resulting in economic investment from the user's point of view. Load balancing or task scheduling in a fog computing environment belongs to a category of NP-hard problem because of the availability of large solution space and it takes a huge time to reach an optimal solution.

Load balancing in fog computing consists of n tasks and m Virtual Machines (VM's). Each task should be processed by each of the m VM's in order to minimize makespan. Scheduling algorithms are used mainly to minimize execution time and execution cost. The Task Scheduling handles the problem in which resources need to be assigned to the received job.

An efficient scheduling algorithm considers balancing the load of the system considering the total execution time of available resources, leading to the minimization of execution cost.

Based on the research gaps outlined following objectives are set for this research work to be accomplished:

1. Identification and analysis of the existing works for efficient load balancing or task scheduling in the Fog Cloud computing environment.
2. Develop algorithms to improve the performance of the resource in Fog Cloud computing
3. To develop a hybrid methodology for efficient task scheduling.
4. Evaluation and comparison of the developed task scheduling techniques.

References

[1] Oussous, A., Benjelloun, F.-Z., AitLahcen, A., &Belfkih, S. (2017). Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences.

[2] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information systems, 47, 98-115.

[3] Zanoon, N., Al-Haj, A., &Khwaldeh, S. M. (2017). Cloud computing and big data is there a relation between the two: a study. International Journal of Applied Engineering Research, 12(17), 6970-6982.

[4] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., ...&Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture, 98, 289-330.

[5] Mostafavi, S., &Shafik, W. (2019). Fog computing architectures, privacy and security solutions. Journal of Communications Technology, Electronics and Computer Science, 24, 1-14.

[6] B. Varghese, N.Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos,``Challenges and opportunities in edge computing," in *Proc. IEEE Int.Conf. Smart Cloud (SmartCloud)*, Nov. 2016, pp. 20_26.

[7] C. Byers, and R. Swanson, "OpenFog reference architecture for Fog computing," OpenFog Consortium White Paper, pp. 1-162, Feb. 2017.

[8] Shanhe Yi, Cheng Li, and Qun Li. 2015. A Survey of Fog Computing: Concepts, Applications and Issues. In Proceedings of the 2015 Workshop on Mobile Big Data (Mobidata '15). Association for Computing Machinery, New York, NY, USA, 37–42.

[9] Dastjerdi, A. V., &Buyya, R. (2016). Fog computing: Helping the Internet of Things realize its potential. *Computer*, *49*(8), 112-116.

[10] Iorgam, M., Feldman, L., Barton, R., Martin, M. J., Goren, N., &Mahmoudi, C. (2018). Fog Computing Conceptual Model, Recommendations of the National Institute of Standards and Technology. In NIST Special publication 500-325.

[11] Li, J., Jin, J., Yuan, D., & Zhang, H. (2017). Virtual fog: A virtualization enabled fog computing framework for Internet of Things. IEEE Internet of Things Journal, 5(1), 121-131.

[12] Prakash, P., Darshaun, K. G., Yaazhlene, P., Ganesh, M. V., &Vasudha, B. (2017). Fog computing: Issues, challenges and future directions. International Journal of Electrical and Computer Engineering, 7(6), 3669.

[13] X.-Q. Pham et al., "A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing," International Journal of Distributed Sensor Networks, vol. 13, no. 11, 2017.

[14] Fersi, G. Fog computing and Internet of Things in one building block: a survey and an overview of interacting technologies. Cluster Comput 24, 2757–2787 (2021).

[15] Caiza, G., Saeteros, M., Oñate, W., & Garcia, M. V. (2020). Fog computing at industrial level, architecture, latency, energy, and security: A review. Heliyon, 6(4), e03706.

[16] Ostberg, P. O., Byrne, J., Casari, P., Eardley, P., Anta, A. F., Forsman, J., Kennedy, J., Le Duc, T., Marino, M. N., & Loomba, R., et al. (2017). Reliable capacity provisioning for distributed cloud/edge/fog computing applications. In In Proceedings of the 2017 European conference on networks and communications (EuCNC) (pp. 1–6). IEEE.

[17] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), 854-864.

[18] Perera, C., Qin, Y., Estrella, J. C., Reiff-Marganiec, S., &Vasilakos, A. V. (2017). Fog computing for sustainable smart cities: A survey. ACM Computing Surveys (CSUR), 50(3), 1-43.

[19] Mekki, T., Jabri, I., Chaari, L., &Rachedi, A. (2020, April). A Survey on Vehicular Fog Computing: Motivation, Architectures, Taxonomy, and Issues. in Proceedings of Workshops of the International Conference on Advanced Information Networking and Applications (pp. 159-168). Springer, Cham.

[20] Deb, K., (2001) "Multi-Objective Optimization Using Evolutionary Algorithms", Wiley, Chichester, UK. Google Scholar Google Scholar Digital Library Digital Library

[21] D. Hoang and T. D. Dang, "FBRC: Optimization of task scheduling in Fog-based Region and Cloud," in Proceedings of the 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 11th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Conference on Embedded Software and Systems, Trustcom/BigDataSE/ICESS 2017, pp. 1109–1114, Australia, August 2017.

[22] Mtshali, M., Kobo, H., Dlamini, S., Adigun, M., &Mudali, P. (2019, August). Multi-objective optimization approach for task scheduling in fog computing. In Proceedings of *2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)* (pp. 1-6). IEEE.

[23] Verma, M., Bhardwaj, N., & Yadav, A. K. (2016). Real time efficient scheduling algorithm for load balancing in fog computing environment. *Int. J. Inf. Technol. Comput. Sci*, *8*(4), 1-10.

[24] D. Hoang and T. D. Dang, ``FBRC: Optimization of task scheduling in fog-based region and cloud," in *Proc. IEEE Trustcom/BigDataSE/ICESS*,Aug. 2017, pp. 1109_1114.

[25] Hussain, M., SaadAlam, M., Sufyan Beg, M. M., & Akhtar, N. (2020). Towards minimizing delay and energy consumption in vehicular fog computing (VFC). *Journal of Intelligent & Fuzzy Systems*, *38*(5), 6549-6560.

[26] Liu, L., Chang, Z., Guo, X., Mao, S., &Ristaniemi, T. (2017). Multiobjective optimization for computation offloading in fog computing. *IEEE Internet of Things Journal*, *5*(1), 283-294.

[27] N. B.V. and R. M. R. Guddeti, "Heuristic-Based IoT Application Modules Placement in the Fog-Cloud Computing Environment," in *Proc. IEEE/ACM International*

Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 24-25,

[28] S. Kabirzadeh, D. Rahbari, and M. Nickray, "A hyper heuristic algorithm for scheduling of fog networks," in *Proc. 21st Conf. Open Innov. Assoc.(FRUCT)*, 2017, pp. 148_155.

[29] EL AMRANI, C., & GIBET TANI, H. (2018). Smarter round robin scheduling algorithm for cloud computing and big data. *Journal of Data Mining & Digital Humanities*.

[30] Singh, S.P., Nayyar, A., Kaur, H., &Singla, A. (2019). Dynamic Task Scheduling using Balanced VM Allocation Policy for Fog Computing Platforms. *Scalable Comput. Pract. Exp., 20*, 433-456.

[31] Ghobaei-Arani, M., Souri, A., &Rahmanian, A. A. (2020). Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*, *18*(1), 1-42.

[32] Agarwal, S., Yadav, S., & Yadav, A. K. (2016). An efficient architecture and algorithm for resource provisioning in fog computing. *International Journal of Information Engineering and Electronic Business*, *8*(1), 48.

[33] Awaisi, K. S., Hussain, S., Ahmed, M., Khan, A. A., & Ahmed, G. (2020). Leveraging IoT and fog computing in healthcare systems. *IEEE Internet of Things Magazine*, *3*(2), 52-56.

[34] Abdel-Basset, M., Mohamed, R., Elhoseny, M., Bashir, A. K., Jolfaei, A., & Kumar, N. (2020). Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications. *IEEE Transactions on Industrial Informatics*, *17*(7), 5068-5076.

[35] Debe, M., Salah, K., Rehman, M. H. U., &Svetinovic, D. (2019). IoT public fog nodes reputation system: A decentralized solution using Ethereum blockchain. IEEE Access, 7, 178082-178093.

[36] Rafique, H., Shah, M. A., Islam, S. U., Maqsood, T., Khan, S., & Maple, C. (2019). A novel bio-inspired hybrid algorithm (NBIHA) for efficient resource management in fog computing. *IEEE Access*, *7*, 115760-115773.